

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10-83293

(43) 公開日 平成10年(1998)3月31日

(51) Int. Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/06	5 4 0		G 0 6 F 9/06	5 4 0 B
	4 1 0			4 1 0 H

審査請求 未請求 請求項の数 15

O L

(全 10 頁)

(21) 出願番号 特願平9-76916

(22) 出願日 平成9年(1997)3月28日

(31) 優先権主張番号 08/625376

(32) 優先日 1996年4月1日

(33) 優先権主張国 米国 (U S)

(71) 出願人 594170738

サン マイクロシステムズ インコーポレ
イテッド

アメリカ合衆国 カリフォルニア州 940

43 マウンテン ヴィュー ガルシア ア
ヴェニュー 2550

(72) 発明者 ウォルター ニールセン

アメリカ合衆国 カリフォルニア州 944

04 フォスター シティ エッジウオー
ター ブールヴァード 9696 アパートメ
ント 704

(74) 代理人 弁理士 中村 稔 (外6名)

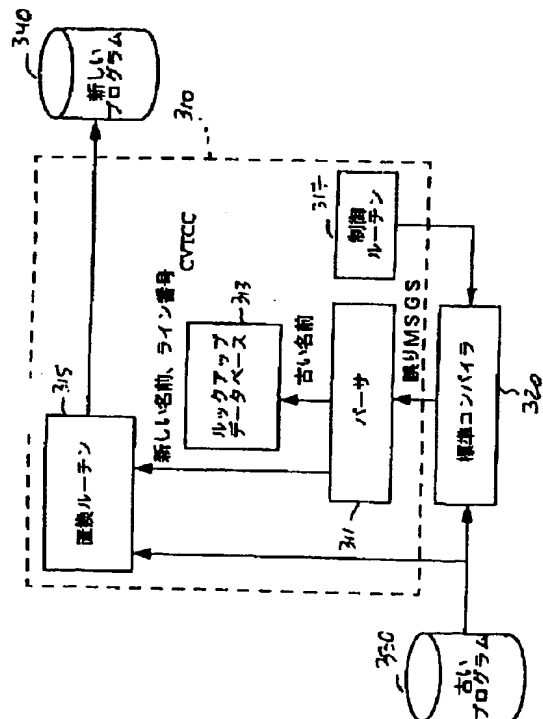
最終頁に続く

(54) 【発明の名称】 誤り訂正コンパイラ

(57) 【要約】

【課題】 ソフトウェア変換を自動化することができる
ツールを備えたシステム及び方法を提供する。

【解決手段】 一つのオペレーティング環境から別のオペ
レーティング環境へのソフトウェア・プログラムの自
動化された変換を容易にするソフトウェア変換ツール。
より特定的には、コンピュータ・プログラムは、コンピ
ュータ・プログラムによって想定されたソフトウェア環
境以外のソフトウェア環境に対するコンパイラを用いて
コンパイルされる。結果として、コンパイラは、誤りメ
ッセージを生成し、かつ誤りメッセージに応じて、コン
ピュータ・プログラム内のソース・コードは、誤りをも
たらしめている条件を除去するために自動的に変更され
る。ソフトウェア変換ツールは、誤り訂正コンパイラ、
即ち、誤りを検出しかつそれらをユーザに与えるだけの
代わりに、それらがリコンパイルーション中に発生しな
いように実際に誤りを訂正することができるコンパイラ
を結果として形成すべく、標準コンパイラでありうる、
コンパイラと共同で動作する。



【特許請求の範囲】

【請求項1】 第2の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第1のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピュータ・インプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備することを特徴とするコンピュータ・インプリメンティッド方法。

【請求項2】 第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備することを特徴とする請求項1に記載の方法。

【請求項3】 請求項2に記載の段階を複数回繰り返すことを具備する請求項2に記載の方法。

【請求項4】 異なるプログラム識別子がデータベースに含まれるような各未知識別子が、その異なるプログラム識別子によって置換されるまで請求項2に記載の段階を複数回繰り返すことを具備する請求項3に記載の方法。

【請求項5】 コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで請求項2に記載の段階を複数回繰り返すことを具備する請求項3に記載の方法。

【請求項6】 コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパiling・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備することを特徴とするコンピュータ・インプリメンティッド方法。

【請求項7】 コンピュータ・プログラムをコンパイルする前に、コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルする段階を更に具備することを特徴とする

請求項6に記載の方法。

【請求項8】 変更する段階は、：誤りメッセージに応じて、未知プログラム識別子に対するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換することを具備することを特徴とする請求項7に記載の方法。

【請求項9】 第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備することを特徴とする請求項8に記載の方法。

【請求項10】 請求項9に記載の段階を複数回繰り返すことを具備する請求項9に記載の方法。

【請求項11】 コンパイラが誤りメッセージを生成しなくなるまで請求項9に記載の段階を繰り返すことを具備する請求項10に記載の方法。

【請求項12】 コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで請求項9に記載の段階を繰り返すことを具備する請求項10に記載の方法。

【請求項13】 第1のソフトウェア環境に対して書かれたプログラムを第2の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第2の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プログラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に応答する置換コードを備えていることを特徴とするコンピュータ・ソフトウェア・システム。

【請求項14】 それにより変換されたプログラムを生成すべくシステムを反復的に置換及びリコンパイルさせるコンパイラ、パーサ、データベース、及び置換コードに結合された制御コードを更に備えていることを特徴とする請求項13に記載の装置。

【請求項15】 その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラ

ム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えていることを特徴とするコンピュータ読取り可能コード。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ソフトウェア開発ツールに関し、特にソフトウェア変換ツールに関する。

【0002】

【従来の技術】ソフトウェア開発は、時間が掛かりかつ面倒な処理である。先に書かれたソフトウェアを保守し、かつ再加工することは、少なくとも同様に面倒である。幾度も、一つのソフトウェア環境において走るべく書かれたプログラムを別の異なるソフトウェア環境において走るフォームに変換するための必要性が生じる。そのような変換は、古いシステムを参照するプログラムにおける全ての言語構成要素を見出し、かつ新しいシステムを参照する言語構成要素でそれらを置換することを必要とする。プログラムが大きく、数十万または百万のコードのラインを含んでいるならば、手動でそのような変換を実行することは、膨大なタスクである。明らかに、そのような変換を実行する機械援助型方法を工夫することが望ましい。

【0003】

【発明が解決しようとする課題】残念ながら、特定テキスト・ストリングの全ての発生が異なる特定テキスト・ストリングで置換されるような、簡単な“探索及び置換”戦略は、有用な結果を供給するにはあまりにも簡単すぎる。自然言語におけるようにコンピュータ言語において、特定言語構成要素の意味は、かなりコンテキスト-依存性である。時々、例えば、ネームのローカル・バージョンが現れる。ネームのローカル・バージョンを汎用の新しいネームで置換することは、走らないかまたは正しく走らないプログラムを結果として生ずる。従って、上記した型のソフトウェア変換を、可能な限り、自動化するソフトウェア・ツールに対する必要性が存在する。

【0004】本発明の目的は、上記従来の技術の必要性に応じて、ソフトウェア変換を自動化することができるツールを備えたシステム及び方法を提供することである。

【0005】

【課題を解決するための手段】本発明の上記目的は、第2の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第1のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピュータ・インプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用い

られるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備するコンピュータ・インプリメンティッド方法によって達成される。

【0006】本発明の方法では、第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備するようにしてもよい。本発明の方法では、上記記載の段階を複数回繰り返すことを具備するようにしてもよい。本発明の方法では、異なるプログラム識別子がデータベースに含まれるような各未知識別子が、その異なるプログラム識別子によって置換されるまで上記記載の段階を複数回繰り返すことを具備するようにしてもよい。本発明の方法では、コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで上記記載の段階を複数回繰り返すことを具備するようにしてもよい。

【0007】また、本発明の上記目的は、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパiling・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備するコンピュータ・インプリメンティッド方法によって達成される。本発明の方法では、コンピュータ・プログラムをコンパイルする前に、コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルする段階を更に具備するようにしてもよい。

【0008】本発明の方法では、変更する段階は、：誤りメッセージに応じて、未知プログラム識別子に対するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換することを具備するようにしてもよい。本発明の方法では、第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをリコンパイルする更なる段階を具備するようにしてもよい。本発明の方法では、上記記載の段階を複数回

繰り返すことを具備するようにしてもよい。本発明の方法では、コンパイラが誤りメッセージを生成しなくなるまで上記記載の段階を繰り返すことを具備するようにしてもよい。

【0009】本発明の方法では、コンパイラによって生成された誤りメッセージに対するデータベースにおいてエントリが見出されなくなるまで上記記載の段階を繰り返すことを具備するようにしてもよい。更に、本発明の上記目的は、第1のソフトウェア環境に対して書かれたプログラムを第2の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第2の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プログラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に応答する置換コードを備えているコンピュータ・ソフトウェア・システムによって達成される。

【0010】本発明のシステムでは、それにより変換されたプログラムを生成すべくシステムを反復的に置換及びリコンパイルさせるコンパイラ、パーサ、データベース、及び置換コードに結合された制御コードを更に備えてもよい。本発明の上記目的は、その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えているコンピュータ読取り可能コードによって達成される。

【0011】

【作用】本発明は、一般的に言えば、一つのオペレーティング環境から別のオペレーティング環境へのソフトウェア・プログラムの自動化された変換を容易にするソフトウェア変換ツールを提供する。本発明の一実施例によれば、コンピュータ・プログラムは、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンパイル

される。結果として、コンパイラは、誤りメッセージを生成する。誤りメッセージに応じて、コンピュータ・プログラム内のソース・コードは、誤りをもたらし条件を除去するために自動的に変更される。ソフトウェア変換ツールは、誤り訂正コンパイラ、即ち、誤りを検出しかつユーザにそれらを与えるだけの代わりに、それらがリコンパイレーション中に発生しないように誤りを実際に訂正することができるコンパイラを結果として形成すべく、標準コンパイラでありうる、コンパイラと共同して動作する。

【0012】本発明は、添付した図面に関する以下の説明から更に理解されうる。

【0013】

【実施例】誤り訂正コンパイラが用いられうるコンピュータ・システムがまず説明され、より特定の誤り訂正コンパイラそれ自体の説明が続く。図1をまず参照すると、誤り訂正コンパイラは、モニタ110、キーボード120及びマウス130を有している、図示された型のコンピュータ・ワークステーションで用いられうる。マウス・カーソル140は、モニタ画面上に現れかつマウスが動かされるように動く。図1のコンピュータ・ワークステーションの内部ハードウェアは、図2に示すような簡略化されたブロック図の形で表されうる。コンピュータ・システムは、中央処理装置202、システム・メモリ204、ディスプレイ206、プリンタ208、キーボード210、ポインティング・デバイス212、ディスク記憶サブシステム214、I/Oコントローラ216、及びシステム・バスのような相互接続手段218を有する。

【0014】本発明のハードウェア環境を説明したので、誤り訂正コンパイラのソフトウェア・コンポーネント及びそれらの相関関係をより具体的に図3を参照してここで説明する。誤り訂正コンパイラは、実際にはソフトウェア変換プログラムと共同で動作している標準コンパイラである。一実施例では、ソフトウェア変換プログラムは、一つのアプリケーション・インターフェイス（例えば、NextStep）とコンパチブルなオブジェクト型Cプログラムを別のアプリケーション・インターフェイス（例えば、OpenStep）とコンパチブルなオブジェクト型Cプログラムに変換するために用いられ、かつオブジェクト型Cの一つのバージョンから別のバージョンにコードを変換するプログラムによって実行される機能を示している、“cvtcc”と名前が付けられる。

【0015】図3を参照すると、変換プログラム310は、4つの主コンポーネント、単純構文解析系（シンブル・パーサ）311、データベース313、置換(substitution)プログラム315及び制御プログラム317を有する。変換プログラム310は、古いソフトウェア環境を想定する古いプログラム330を、新しいソフトウェア環境を想定する新しいプログラム340の中に変換

すべく標準コンパイラ320と共同で動作する。パーサ311及び置換プログラム315は、アクセス・プログラムと一緒に機能する。データベース313及びアクセス・プログラムは、古い言語構成要素を新しい言語構成要素の中にマップする。特に、パーサ311は、コンパイラ20から誤りメッセージを受け取りかつ古い資源名及びファイル名及び古い資源名が現れるライン番号を誤りメッセージから抽出する。古い名前は、データベース313における対応している新しい名前を調べるために用いられる。新しい名前、ファイル名及びライン番号は、次いで、古い名前に対する新しい名前の実際の置換を実行する、置換プログラム315に供給される。データベースは、古いソフトウェア環境及び新しいソフトウェア環境の両方に慣れている有能なプログラマによって手動で構築される。

【0016】古いプログラムを新しいプログラムに変換するために、変換プログラム310は、制御プログラム317によって繰り返し走らされる。まず、ユーザは、古いプログラムをコンパイルするために、古いシステム環境の代わりに新しいシステム環境にコンパイルすべく設計または構成された、コンパイラを呼出す。結果として、古いシステム言語構成要素の使用で誤りが生成される。制御プログラムの制御下で、誤り訂正コンパイラ（即ち、コンパイラ320と共同する変換プログラム310）は、一つ以上の古いシステム構成要素を変換データベースにおいて見出される対応している新しい構成要素で繰り返し置換しかつリコンパイルする。それゆえに、誤り訂正コンパイラは、それ以上の誤りをフィックスすることができなくなるまで合成誤りをフィックスするような、実際のコンパイラを呼出し続ける。

【0017】より特定の図4を参照すると、誤り訂正コンパイラが走り始めるとき、フラグfixedSome（フィックスド・サム）は、false（フォールス）に設定される（ステップ401）。レギュラ・コンパイラは、次いで、変換されるべきプログラムをコンパイルすべく呼出され、かつコンパイラからの誤りリスティングが捕捉される（ステップ403）。次いで、最初の誤りメッセージがパースされる（ステップ405）。誤りが認識され

る型のものではないならば（ステップ407）、そして次の誤りメッセージが存在するならば（ステップ409）、それは、パースされる、等。誤りが認識される型、即ち、“identifier not found”（識別子が見出されなかった）メッセージのような、誤り訂正コンパイラがフィックスすることが可能でありうる型のものであれば（ステップ407）、古いプログラムの構成要素の名前は、古い名前への参照の位置（ファイル、ライン番号）と一緒に、誤りメッセージから抽出される（ステップ411）。古い名前は、次いで、データベースにおいて調べられる（ステップ413）。古い名前がデータベースにおいて見出されなければ（ステップ415）、次の誤りメッセージ（あれば）が処理される。古い名前がデータベースにおいて見出されるならば（ステップ415）、古い名前は、データベースにおいて見出される新しい名前で置換され（ステップ417）、その後フラグfixedSomeは、真に設定される（ステップ419）。

【0018】上記オペレーションのシーケンスは、次の誤りメッセージが存在しなくなるまで誤りメッセージの全てに対して実行される（ステップ409）。次いで、fixedSomeが真であれば（ステップ421）、処理は、fixedSomeをfalseにリセットし（ステップ401）かつ変更されたプログラムをリコンパイルする（ステップ403）ことによって、始めから繰り返し替えされる。オペレーションは、誤り訂正コンパイラがそれ以上の誤りをフィックスすることができなくなるまで、即ち、ステップ421でfixedSomeがfalseに留まるまでこのファッションで進行する。次いで誤り訂正コンパイラは、停止する。誤り訂正コンパイラがフィックスすることができなかった誤りは、次いで、手動でフィックスされうる。

【0019】誤り訂正コンパイラは、次に示す添付資料からさらに理解されうる。表1に示す添付資料Aは、サンプル・データベースのリスティングである。表2に示す添付資料Bは、誤り訂正コンパイラの使用中に生成された画面プリントアウトである。

【0020】

【表1】

```

@implementation Menu
    AppDelegate *AppDelegate;
    AppDelegate *AppDelegate;
    @interface Menu : Panel
    @interface NSMenu : NSPanel
    + setMenuIcon:(NSImage *)aIcon;
    + (void) setMenuIcon:(NSImage *)aIcon;
    + (NSImage *)menuIcon;
    + (NSImage *)menuIcon;
    @- init;

    - initWithTitle:(const char *)aTitle;
    - (id) initWithTitle:(NSString *)aTitle;
    // is aTitle is NULL, change it to ""

    - (id) initWithTitle:(NSString *)aTitle action:(SEL)aSelector keyEquivalent:(NSString *)
    - addItem:(const char *)aTitle action:(SEL)aSelector keyEquivalent:(unsigned short)charCo
    - (id) initWithTitle:(NSString *)aTitle action:(SEL)aSelector keyEquivalent:(NSString *)
    - setSubmenu:(NSMenu *)aSubmenu;
    - (NSMenuItem *) setSubmenu:(NSMenu *)aMenu submenu:(NSMenuItem *)aCell;

    - (NSMenu *)superMenu;

    - (NSMenu *)attachedMenu;

    - (BOOL) isAttached;

    - (BOOL) isDestroyed;
    - dealloc;
    - (NSMenuItem *) itemAtIndex;
    - setItemAtIndex:(int)anIndex;
    - (void) setItemAtIndex:(NSMenuItem *)anItem;
    - display;

    - setFrame;
    - (void) setFrame;

    - windowMoved:(NSEvent *)theEvent;

    - close;

    - update;

    - setAutoUpdate:(BOOL)flag;
    - (void) setAutoUpdate:(BOOL)flag;

    - (BOOL) autoUpdate;
    - findCellWithTag:(int)tag;
    - (id) cellWithTag:(int)tag;
    - getLocation:(NSEvent *)theEvent forSubmenu:(NSMenu *)aSubmenu;
    - (NSEvent *) locationForSubmenu:(NSMenu *)aSubmenu;
    - mouseDown:(NSEvent *)theEvent;

    - rightMouseDown:(NSEvent *)theEvent;

    @- new;
    - initWithTitle:(const char *)aTitle;

    @end
    @end
    @interface Menu (SubmenuDummyAction)
    @interface NSMenu (NSSubmenuDummyAction)
    - submenuAction:(id) sender;
    - (void) submenuAction:(id) sender;
    @end
    @end

    @interface NSMenuItem (NSMenuItemActionResponder)
    - (BOOL) validateCell:(id) aCell;
    @end

```

```

>>:271 more /all.h
1
2 #import <AppKit/AppKit.h>
3
4 void foo() {
5     Menu *fired;
6     (fired initWithTitle: "abc");
7 }
8
9 >>:272 cc -o /all.h -c -I/opt/csc/export/obj/dev8/obj/asm/include \
10 -I/usr/openwin/include/X11 -I/usr/openwin/include
11 "/home/rob/all.h", line 2: Error: Could not open include file <AppKit/AppKit.h>.
12 "/home/rob/all.h", line 5: Error: Menu is not defined.
13 "/home/rob/all.h", line 5: Error: fired is not defined.
14 "/home/rob/all.h", line 6: Error: fired is not defined.
15 4 Errors! Detected.
16 >>:273 bin/ovcc -o /all.h -c -I/opt/csc/export/obj/dev8/obj/asm/include \
17 -I/usr/openwin/include/X11 -I/usr/openwin/include
18 Using default filebase: /home/rob/nyana/asm/ConversionTools/appkit.h
19 ...
20 ... Compiling ...
21 ... Attempting to fix errors ...
22 ... Fixed these bugs ...
23 "/home/rob/all.h", line 2: Error: Could not open include file <AppKit/AppKit.h>.
24   -> from source: 'AppKit/AppKit.h'
25   -> source: 'AppKit/AppKit.h'
26 ...
27 ... Fixed some missing include file errors: recompiling ...
28 ... Attempting to fix errors ...
29 ... Fixed these bugs ...
30 "/home/rob/all.h", line 5: Error: Menu is not defined.
31   -> from file=appkit/Menu.h, source= 'Interface Menu : Panel'
32   -> file=AppKit/NSMenu.h, source= 'Interface NSMenu : NSTPanel'
33 ...
34 ... Fixed some errors, recompiling ...
35 ... Attempting to fix errors ...
36 ... Fixed these bugs ...
37 "/home/rob/all.h", line 6: Warning: Cannot find instance method initWithTitle: \
38   for class NSTMenu:NSPanel:NSWindow:NSResponder:NSObject.
39   -> from file=appkit/Menu.h, source= '- initWithTitle:(const char *)aTitle;'
40   -> file=AppKit/NSMenu.h, source= '- (id)initWithTitle:(NSString *)aTitle;'
41   / if aTitle is NULL, change it to 3"
42 ...
43 ... Fixed some errors, recompiling ...
44 ... Attempting to fix errors ...
45 ... Fixed all we can in this file ...
46
47 >>:274 more /all.h
48
49 #import <AppKit/AppKit.h>
50
51 void foo() {
52     NSMenu *fired;
53     (fired initWithTitle: "abc");
54 }

```

添付資料Aを参照すると、説明された実施例における、データベースは、非常に簡単な構成である。最初のテキスト・カラムで始まっているエントリは、置換されるべき古い名前である。置換エントリがデータベースにおいて見出されるならば、それは、第2のテキスト・カラムで始まっている、直後のラインに入力される。全ての古い名前が対応している新しい名前をもっているわけではなく、その逆もそうである。記号“#”、“@”、“+”、“-”は、オブジェクトCにおいて認識された意味を有する。添付資料Bを参照すると、誤り訂正コンパイラのオペレーションの具体的な例を供給するために、SparcStation（スパークステーション）・コンピュータで走っておりかつSunOS（サンオーエス）オペレーティング・システム下で走っている誤り訂正コンパイラのバージョンは、ライン1～7の例示プログラムに呼出された。ライン8から始まって、コンパイラが走らせ、ライン11～14にリストされた4つの誤りメッセージを結果として生ずる。ライン16から始まって、変

換プログラムは、コンパイラによって生成された誤りメッセージ・ファイルを用いて走らせる。変換プログラムは、組込みファイル誤りをフィックスすることをまず試みる。そこでライン21～24において、変換プログラムは、それが、データベースにおいて特定されたように組込みファイルの名前を変更したことを報告する。コンパイラは、次いで、変更されたプログラムをリコンパイルすべく再び呼出される（ライン26）。更なる誤りメッセージに報告された誤りは、次いで、フィックスされる。ライン44では、誤り訂正コンパイラは、可能な限り全ての誤りをフィックスしたことを報告する。合成変更プログラムは、ライン48～54にリストされる。オリジナル・プログラムの“Menu”は、“NSMenu”に変更され、かつ“initWithTitle”は、“initWithTitle”に変更されたということに注目する。

【0022】本発明は、その精神または重要な特徴から逸脱することなく他の特定のフォームで実施することができるといことが当業者によって理解されるである

う。上述の記載は、従って全ての関して説明のためであり限定的でなものである。本発明の範疇は、特許請求の範囲によって示され、かつその同等物の意味及び範囲内にある全ての変更は、その中に含まれることを意図する。

【0023】

【発明の効果】本発明のコンピュータインプリメンティッド方法は、第2の、異なるソフトウェア環境で走る変更されたコンピュータ・プログラムを生成するために第1のソフトウェア環境で走るべく書かれたコンピュータ・プログラムを変更するコンピュータインプリメンティッド方法であって：コンピュータ・プログラムに現されるプログラム識別子及び変更されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含している辞書データベースをアセンブルし；第2の異なるソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイラは、未知のプログラム識別子に出会ったとき、未知のプログラム識別子を含んでいる誤りメッセージを生成し；誤りメッセージに応じて、それで未知のプログラム識別子を置換するエントリをデータベースから検索し；かつエントリに含まれた異なるプログラム識別子でコンピュータ・プログラムの未知のプログラム識別子の少なくとも一例を置換する段階を具備するので、ソフトウェア変換を可能な限り自動化することができる。

【0024】本発明のコンピュータインプリメンティッド方法は、コンピュータ・プログラムによって想定されたソフトウェア環境以外のソフトウェア環境に対するコンパイラを用いてコンピュータ・プログラムをコンパイルし、コンパイル・オペレーションは、誤りメッセージを生成し；かつ誤りメッセージに応じて、誤りをもたらしている条件を除去するためにコンピュータ・プログラム内のソース・コードを自動的に変更する段階を具備するので、ソフトウェア変換を可能な限り自動化することができる。本発明のコンピュータ・ソフトウェア・システムは、第1のソフトウェア環境に対して書かれたプログラムを第2の、異なるソフトウェア環境に対する変換されたプログラムに変換するコンピュータ・ソフトウェア・システムであって：未知プログラム識別子がプログラムで出会うときに誤りメッセージを生成する、第2の、異なるソフトウェア環境に対するコンパイラ；コンパイラから誤りメッセージを受取りかつ未知プロ

ラム識別子及びプログラム内の未知プログラム識別子の位置を出力するパーサ；未知プログラム識別子及び変換されたコンピュータ・プログラムで用いられるべき異なるプログラム識別子を含んでいるデータベース・エントリを包含しており、未知プログラム識別子を受取りかつ異なるプログラム識別子をそれに応じて出力する辞書データベース；及びコンピュータ・プログラム内の異なるプログラム識別子で未知プログラム識別子を置換するために未知プログラム識別子の位置及び異なるプログラム識別子に回答する置換コードを備えているので、ソフトウェア変換を可能な限り自動化することができる。

【0025】本発明のコンピュータ読取り可能コードは、その中に構成されたコンピュータ読取り可能プログラム・コードを有しているコンピュータ使用可能媒体を含んでいる製造物における、コンピュータ読取り可能コードであって：コンパイラからの誤りメッセージに応じて、コンパイラにより知られていない未知プログラム識別子に対するエントリをデータベースから検索すべく構成されるコンピュータ読取り可能プログラム・コード；及びデータベースのエントリに包含された異なるプログラム識別子で未知プログラム識別子の少なくとも一例を置換すべく構成されたコンピュータ読取り可能プログラム・コードを備えているので、ソフトウェア変換を可能な限り自動化することができる。

【図面の簡単な説明】

【図1】本発明が用いられうる通常のコンピュータ・ワークステーションの斜視図である；

【図2】図1のコンピュータ・ワークステーションの一部のブロック図である；

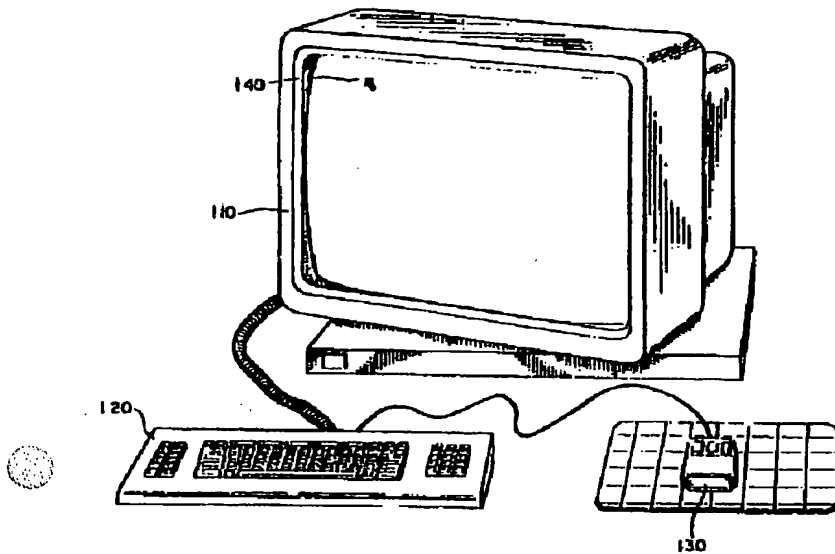
【図3】本発明の誤り訂正コンパイラのブロック図である；

【図4】図3の誤り訂正コンパイラのオペレーションを説明するフロー図である。

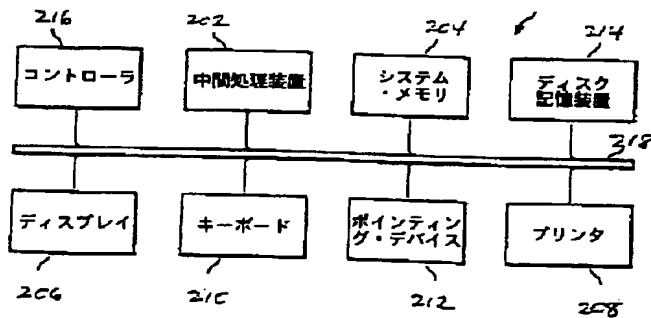
【符号の説明】

- 310 変換プログラム
- 311 単純構文解析系（シンプル・パーサ）
- 313 データベース
- 315 置換プログラム
- 317 制御プログラム
- 320 標準コンパイラ
- 330 古いプログラム
- 340 新しいプログラム

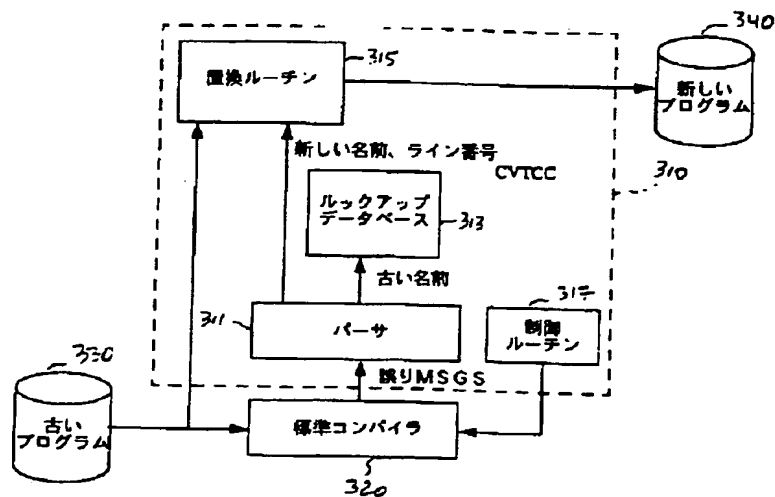
【図1】



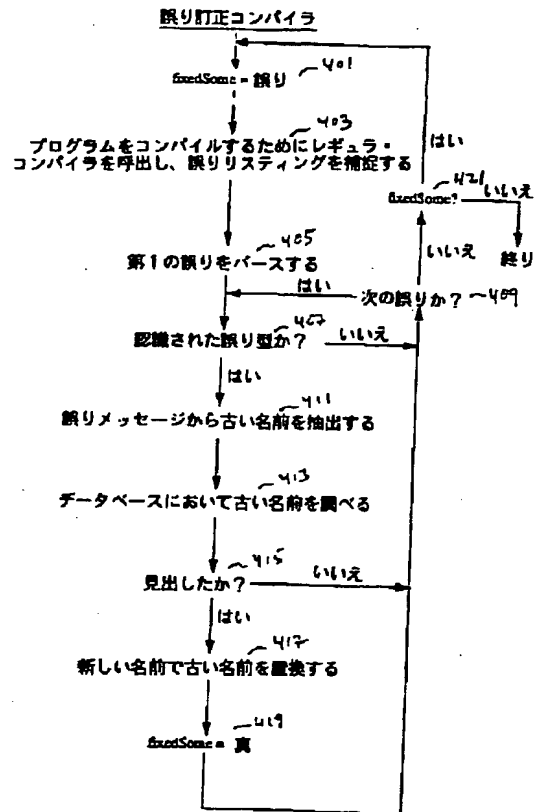
【図2】



【図3】



【図4】



フロントページの続き

(72)発明者 ジェームズ ホームルンド
アメリカ合衆国 カリフォルニア州
94303 パロ アルト アーストワイルド
コート 73